Alibaba Cloud
MQTT

Comparison Between MQTT and RocketMQ

Issue: 20190524

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified , reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates . The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd / d C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 Scenario comparison

This topic describes the association and differences between AliwareMQ for IoT and traditional MOM based on #unique_4, and provides model selection recommendations in actual scenarios.

Background

Traditional MOM, such as RocketMQ and Kafka, is intended for microservices and big data and implements message storage and forwarding. The message producer and consumer are applications for user services deployed on ECS instances.

Traditional MOM is applicable in scenarios where user services deployed on ECS instances adopt fixed technology stacks and language platforms. However, traditional MOM cannot deal with scenarios with access by massive multi-platform devices developed in multiple languages, where service properties play an important role during message production and consumption. Examples are the mobile Internet and IoT scenarios.

Designed based on the single-responsibility principle, AliwareMQ for IoT is a stateless gateway intended for the mobile Internet and IoT, focusing only on the access, management, and message transmission of massive mobile devices. Messages to be stored are routed to backend storage services, such as RocketMQ and Kafka.

Based on such a division of responsibilities, AliwareMQ for IoT routes the messages sent by devices to the bound storage service. Off-premises applications can still use traditional microservice development solutions and interact with terminal devices through an off-premises storage service. AliwareMQ for IoT enables data exchange between off-premises applications and devices.

Scenario comparison

A service scenario may include different types of application components, each of which plays a different role. Therefore, when selecting a message service, you need to understand the association and differences between AliwareMQ for IoT and traditional MOM and use them in combination properly. For example, component A uses AliwareMQ for IoT to send and receive messages, whereas component B uses RocketMQ for the same purpose.

The following describes the differences between AliwareMQ for IoT and traditional MOM based on scenarios. RocketMQ is used as an example for comparison. Other message services, such as Kafka and AMQP (RabbitMQ), observe the same rules.

Table 1-1: Scenario comparison

Service	Scenarios
AliwareMQ for IoT	AliwareMQ for IoT is applicable in mobility scenarios with access by massive devices, each of which maintains a relatively small data volume. Therefore, AliwareMQ for IoT can be used to process messages transmitted by a large number of online MQTT clients, each of which maintains a relatively small message volume. For example, many enterprises operate tens of thousands of and even millions of devices.
RocketMQ	RocketMQ is a messaging engine that is oriented towards user services deployed on ECS instances and mainly used for decoupling, asynchronous notification, and load shifting between service components. It is applicable in scenarios with a relatively small number of ECS instances that need to process massive messages and require high throughput. In general, only a few enterprises operate more than 10,000 ECS instances. Therefore, RocketMQ can be used to assist servers with massive data processing and analysis.

Scenarios of use of AliwareMQ for IoT and RocketMQ together

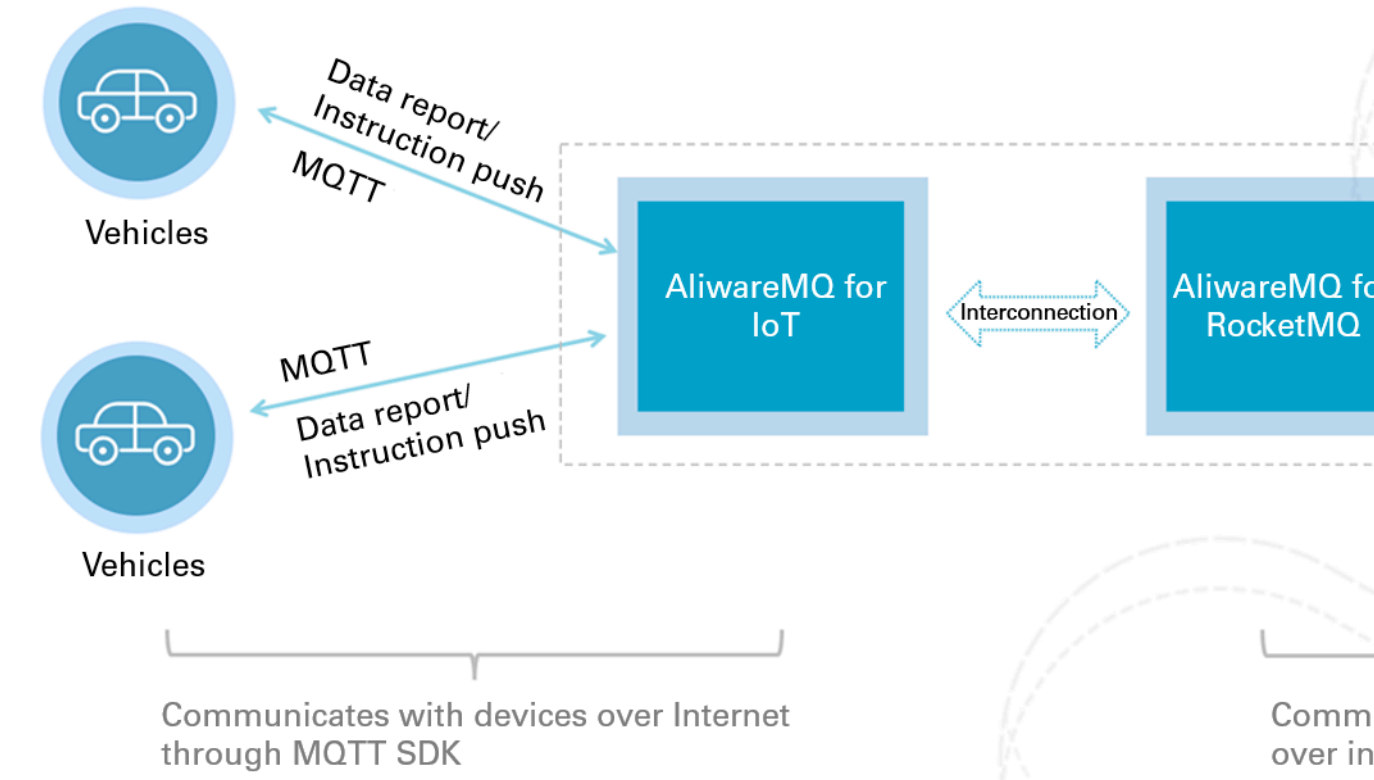
- Scenario 1

  On the IoT, tens of thousands of and even millions of sensors can use AliwareMQ for IoT to upload data, while applications for user services deployed on ECS instances can use RocketMQ to analyze and process the data.

- Scenario 2

  On the Internet of Vehicles (IoV), you may need to upload vehicle information about millions of vehicles to the cloud (ECS instances), while the cloud delivers commands to any specific vehicles or broadcasts commands to all vehicles. Vehicles can connect to AliwareMQ for IoT through the MQTT SDK for uploading data and receiving commands. Monitoring and management systems (data analysis systems) can use the RocketMQ SDK to subscribe to messages and deliver commands. The following figure shows the details.

Figure 1-1: Scenario



Based on the preceding differences, we recommend that you use AliwareMQ for IoT for mobile devices and use RocketMQ (or other message services) for applications on user services that are deployed on ECS instances.

Feature comparison

The following table compares the features of AliwareMQ for IoT and RocketMQ.

Table 1-2: Feature comparison

Feature	AliwareMQ for IoT	RocketMQ
Client connections	AliwareMQ for IoT supports message processing for massive clients, which reach millions or tens of millions in quantity.	RocketMQ supports message processing for a relatively small number of servers, generally less than 10,000 in quantity.
Message volume per client	Each MQTT client processes a small number of messages, and sends and receives messages at regular intervals.	Each MQTT client processes a large number of messages and requires high throughput.
Deployment	Mobile devices, app software, and H5 pages	Server-side applications
Consumption mode	Broadcasting consumption	*Clustering consumption and broadcasting consumption*
Sequence	Messages can be sent in order but cannot be received in order (which will be available in the future).	Messages can be sent and received in order.
Multi-language/system support (TCP)	Java, C, C++, . NET, Android, iOS, Python, JS, and Go	Java, C++, and . NET
Access credential	Permanent RAM access and on-demand access using an MQTT token. For more information, see *Authentication overview*.	*Permanent RAM access* and *on-demand STS-authorized access*

Model selection instructions

The general principle of model selection is as follows:

- RocketMQ is recommended for applications for user services deployed on ECS instances.
- AliwareMQ for IoT is recommended for applications that are deployed on mobile devices, apps, or web pages.

The following table lists the selection recommendations on AliwareMQ for IoT and RocketMQ in common scenarios.

Table 1-3: Recommended model

Scenario	Deployment	AliwareMQ for IoT	RocketMQ
Status data reporting by devices	Mobile device	√	×
Receiving, processing, and analysis of device-reported data	Mobile device	×	√
Delivery of control commands to multiple devices	Server	×	√
Live broadcasting, bullet screen, chat apps that require sending and receiving messages	Application	√	×
Receiving and analysis of chat messages on the MQTT broker	Server	×	√

Note:

√ indicates that the MQ service is recommended, whereas × indicates that the MQ service is not recommended.

2 Message structure mappings

AliwareMQ for IoT is a gateway product that is intended for mobile devices. It is used for sending and receiving messages either independently or with other message storage services, such as RocketMQ.

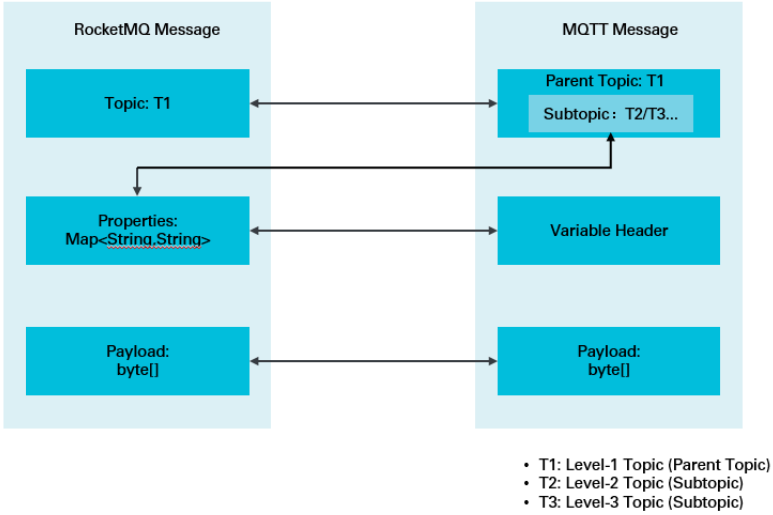
This topic describes the mappings between the message structures and properties involved in interaction with AliwareMQ for IoT by using the RocketMQ SDK, helping you better understand and use AliwareMQ for IoT and RocketMQ.

If you use AliwareMQ for IoT independently, ignore the mappings described in this topic. Just observe the MQTT protocol.

For more information about AliwareMQ for IoT, see *What is AliwareMQ for IoT?* and *Terms*.

Message structure mappings

AliwareMQ for IoT and RocketMQ are messaging systems that are based on the publish-subscribe model and have similar concepts. The following figure shows the differences in the major concepts and mappings between AliwareMQ for IoT and RocketMQ.



As shown in the preceding figure, AliwareMQ for IoT supports multi-level topics, whereas RocketMQ supports one-level topics. Therefore, a level-1 topic in AliwareMQ

for IoT is mapped to a topic in RocketMQ, and level-2 and -3 topics in AliwareMQ for IoT are mapped to the message properties in RocketMQ.

The RocketMQ protocol supports messages with custom properties, whereas the MQTT protocol does not support properties for the moment. However, part of the information in AliwareMQ for IoT is mapped to message properties in RocketMQ. This facilitates tracing of the headers and device information in the MQTT protocol and allows users of the RocketMQ SDK to retrieve such information.

Note:

For information about how to configure mappings from properties in RocketMQ to information in AliwareMQ for IoT, see the table in *Property mappings*.

RocketMQ and AliwareMQ for IoT use the data serialization results of your service messages as the payload, and do not perform further encoding and decoding of the service messages.

Property mappings

The following table lists the property mappings currently supported between AliwareMQ for IoT and RocketMQ. Information can be set or retrieved by reading and writing these properties during interaction between applications that use the SDKs of RocketMQ and AliwareMQ for IoT.

For more information about QoS, cleanSession, topics, and client IDs, see *Terms*.

Property key	Valid value	Description
qoslevel	0, 1, and 2	This property can be set when RocketMQ sends messages to AliwareMQ for IoT. If it is not set, the default value 1 is used. It can be read directly from the messages that AliwareMQ for IoT sends to RocketMQ.

Property key	Valid value	Description
cleansessionflag	true and false	This property can be set when RocketMQ sends P2P messages to MQTT clients. If it is not set, the default value "true" is used. This property is optional for other message types. It can be read directly from the messages that AliwareMQ for IoT sends to RocketMQ.
mqttSecondTopic	A string that indicates a specific subtopic	This property can be set when a subtopic is required to filter the messages that RocketMQ sends to MQTT clients. If it is not set, the default value "null" is used. It can be read directly from the messages that AliwareMQ for IoT sends to RocketMQ.

Property key	Valid value	Description
mqttRealTopic	The sub-level string that services expect message-receiving clients to display	This property can be set when MQTT clients are expected to display the specified subtopic name after receiving messages from RocketMQ. This property is typically applied to P2P messages. If it is not set, P2P messages use an invariable topic name by default. This property is unavailable for the messages that AliwareMQ for IoT sends to RocketMQ.
clientId	A string that indicates a specific client ID	This property cannot be set and is used to trace the ID of the MQTT client that sends a message to RocketMQ.